

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION
ATTORNEY DOCKET NO. 200400479-2

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Keerthi Bhusdhan K. N. et al.
Application No.: 10/827,523
Filing Date: April 20, 2004

Confirmation No.: 2763
Examiner: SMITH, Cheneca
Group Art Unit: 2192

Title: Method and Apparatus for Translating Binary Code

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on December 19, 2009.

- The fee for filing this Appeal Brief is \$540.00 (37 CFR 41.20).
 No Additional Fee Required.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

- (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

1st Month \$130 2nd Month \$490 3rd Month \$1110 4th Month \$1730

- The extension fee has already been filed in this application.
 (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of \$ 540. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

Respectfully submitted,

Keerthi Bhusdhan K. N. et al.

By /Steven L. Nichols/

Steven L. Nichols

Attorney/Agent for Applicant(s)

Reg No. : 40,326

Date : February 18, 2009

Telephone : 801-572-8066

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the Patent Application of
Keerthi Bhushan K N et al.
Application No. 10/827,523
Filed: April 20, 2004
For: Method and Apparatus for Translating
Binary Code

Group Art Unit: 2192

Examiner: SMITH, Cheneca

Confirmation No.: 2763

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an Appeal Brief under Rule 41.37 appealing the decision of the Primary Examiner dated October 20, 2008 (the “final Office Action” or “Action”). Each of the topics required by Rule 41.37 is presented herewith and is labeled appropriately.

I. Real Party in Interest

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249 Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

II. Related Appeals and Interferences

There are no appeals or interferences related to the present application of which the Appellant is aware.

III. Status of Claims

Claims 29, 34 and 35 have been previously cancelled without prejudice or disclaimer.

Claims 1-28 and 30-33 are currently pending in the application and stand finally rejected.

Accordingly, Appellant appeals from the final rejection of claims 1-28 and 30-33, which claims are presented in the Appendix.

IV. Status of Amendments

No amendments have been filed subsequent to the final Office Action of October 20, 2008, from which Appellant takes this appeal.

V. Summary of Claimed Subject Matter

The present application discloses methods, systems, and computer program products for translating binary code instructions from a source format to a target format for processing by a target processor. (*Appellant's specification, abstract and p. 4, lines 6-14*).

The claimed methods, systems, and computer program products are directed to the performance of certain steps, including identifying a source instruction (103) (*Appellant's specification, p. 4, lines 6-14, p. 5, lines 6-16, and p. 6, lines 10-17*); selecting a translation template (109) corresponding to the identified source instruction (103) (*Appellant's specification, p. 4, lines 6-14, p. 5, lines 6-16, and p. 6, lines 10-17*), the template (109) providing a set of target instructions semantically equivalent to the identified source instruction (103) (*Appellant's specification, p. 4, lines 6-14, and p. 5, lines 6-16*). The identified source instruction (103) is then translated in accordance with the template (109) (*Appellant's specification, p. 4, lines 6-14, and p. 5, lines 6-16*), wherein the translating includes converting the source instruction (103) into a source intermediate data structure having a plurality of members (*Appellant's specification, p. 10, line 1 to p. 12, line 16*), mapping the members in the source intermediate data structure to corresponding members in a target intermediate data structure according to the template (109) (*Appellant's specification, p. 11, lines 1-5, p. 13, lines 20-25*), and converting the target intermediate data structure into a target instruction (107) (*Appellant's specification, p. 6, lines 10-17, p. 11, line 25 to p. 12, line 16*). The target instruction (107) is then output for processing by a target processor. (*Appellant's specification, p. 4, lines 6-14, and p. 5, lines 6-16*).

The template-based translation disclosed in the claims overcomes prior art problems in the cost of translation and optimization over execution of translated code, particularly in applications that have poor code locality or frequently cause a translation-cache flush.

(*Applicant's specification*, p. 17, lines 25-30). Particularly, the template-based translation disclosed in the present claims provides a significant reduction in translation overhead without compromising code quality or flexibility to perform optimizations. (*Id.*).

Turning to Appellant's specific claims,

Claim 1 recites:

A method of translating binary code instructions from a source format to a target format for processing by a target processor (*Appellant's specification, abstract and p. 4, lines 6-14*), said method comprising the steps of:

identifying a source instruction (103) (*Appellant's specification, p. 4, lines 6-14, p. 5, lines 6-16, and p. 6, lines 10-17*);

selecting a translation template (109) corresponding to said identified source instruction (103) (*Appellant's specification, p. 4, lines 6-14, p. 5, lines 6-16, and p. 6, lines 10-17*), said template providing a set of target instructions semantically equivalent to said identified source instruction (103) (*Appellant's specification, p. 4, lines 6-14, and p. 5, lines 6-16*);

translating said identified source instruction (103) in accordance with said template (109) (*Appellant's specification, p. 10, line 1 to p. 12, line 16*), wherein said translating comprises:

converting said source instruction (103) into a source intermediate data structure having a plurality of members (*Appellant's specification, p. 10, line 1 to p. 12, line 16*);

mapping said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template (*Appellant's specification, p. 11, lines 1-5, p. 13, lines 20-25*); and

converting said target intermediate data structure into a target instruction (107) (*Appellant's specification, p. 6, lines 10-17, p. 11, line 25 to p. 12, line 16*); and

outputting said target instruction (107) for processing by said target processor (*Appellant's specification, p. 4, lines 6-14, and p. 5, lines 6-16*).

Claim 13 recites:

An apparatus for translating binary code instructions from a source format to a target format for processing by a target processor (*Appellant's specification, abstract and p. 4, lines 6-14*), the apparatus comprising:

at least one processor configured to execute code embodied on a computer readable medium (*Appellant's specification, p. 10, line 1 through p. 12, line 16, p. 17, lines 1-8*);

an instruction identifier (103) embodied within said code for identifying a source instruction (*Appellant's specification, p. 4, lines 6-14, p. 5, lines 6-16, and p. 6, lines 10-17*);

a template selector embodied within said code for selecting a translation template (109) corresponding to said identified source instruction (103) (*Appellant's specification, p. 4, lines 6-14, p. 5, lines 6-16, and p. 6, lines 10-17*), said translation template (109) comprising a set of target instructions semantically equivalent to said identified source instruction (103) and further comprising input and output resources (*Appellant's specification, p. 4, lines 6-14, p. 5, lines 6-16, and Fig. 1*); and

a translator (105) embodied within said code for translating said identified source instruction (103) (*Appellant's specification, p. 4, lines 6-14, and p. 5, lines 6-16*), wherein said translator is configured to

convert said source instruction (103) into a source intermediate data structure having a plurality of members (*Appellant's specification, p. 10, line 1 to p. 12, line 16*);

map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template (*Appellant's specification, p. 11, lines 1-5, p. 13, lines 20-25*); and

convert said target intermediate data structure into a target instruction (107) (*Appellant's specification, p. 6, lines 10-17, p. 11, line 25 to p. 12, line 16*); and

an output buffer embodied within said code for outputting said target instruction (107) for processing by said target processor (*Appellant's specification, p. 4, lines 6-14, and p. 5, lines 6-16*).

Claim 25 recites:

A computer program product for translating binary code instructions from a source format to a target format for processing by a target processor (*Appellant's specification, abstract and p. 4, lines 6-14*), comprising a computer-readable medium (*Appellant's specification, p. 17, lines 1-8*), further comprising:

a template (109) embodied within said computer-readable medium for use in a binary code translator (105) for translating binary code instructions from a source format to a target format for processing by a target processor (*Appellant's specification, p. 5, lines 19-24*), the template (105) comprising:

a template identifier for uniquely associating said template to a source instruction (103) (*Appellant's specification, p. 5, lines 19-24*);

a set of target instructions in a target format semantically equivalent to the source instruction (103) (*Appellant's specification, p. 5, lines 19-24*);

computer usable program code embodied within said computer-readable medium configured to convert said source instruction (103) into a source intermediate data structure having a plurality of members (*Appellant's specification, p. 10, line 1 to p. 12, line 16*);

computer usable program code embodied within said computer-readable medium configured to map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template (109) (*Appellant's specification, p. 11, lines 1-5, p. 13, lines 20-25*); and

computer usable program code embodied within said computer-readable medium configured to convert said target intermediate data structure into a target instruction (107) (*Appellant's specification, p. 6, lines 10-17, p. 11, line 25 to p. 12, line 16*).

Claim 33 recites:

A computer program product for translating binary code instructions from a source format to a target format for processing by a target processor (*Appellant's specification, abstract and p. 4, lines 6-14*), comprising:

a computer-readable medium (*Appellant's specification, p. 17, lines 1-8*), comprising:

a first set of codes for causing a computer to identify a source instruction (103) (*Appellant's specification, p. 4, lines 6-14, p. 5, lines 6-16, and p. 6, lines 10-17*);

a second set of codes for causing a computer to select a translation template (109) corresponding to said identified source instruction (103) (*Appellant's specification, p. 4, lines 6-14, p. 5, lines 6-16, and p. 6, lines 10-17*), said template (109) providing a set of target format instructions semantically equivalent to said identified source instruction (103) (*Appellant's specification, p. 4, lines 6-14, and p. 5, lines 6-16*);

a third set of codes for causing a computer to translate said identified source instruction in accordance with said template (109) (*Appellant's specification, p. 4, lines 6-14, and p. 5, lines 6-16*) by converting said identified source instruction (103) to an intermediate source data structure (*Appellant's specification, p. 10, line 1 to p. 12, line 16*), mapping members of said intermediate source data structure to members of a target intermediate data structure according to said template (109) (*Appellant's specification, p. 11, lines 1-5, p. 13, lines 20-25*), and converting said target intermediate data structure into a target instruction (107) (*Appellant's specification, p. 6, lines 10-17, p. 11, line 25 to p. 12, line 16*); and

a fourth set of codes for causing a computer to output said translated instruction (107) for processing by said target processor (*Appellant's specification, p. 4, lines 6-14, and p. 5, lines 6-16*).

VI. Grounds of Rejection to be Reviewed on Appeal

The final Office Action raised the following grounds of rejection.

(1) Claims 25-28 and 30-32 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

(2) Claims 1-5, 7-17, 19-28, and 30-33 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,075,937 to Scalzi et al. (“Scalzi”) in view of U.S. Patent No. 5,894,576 to Bharadwaj (“Bharadwaj”).

(3) Claims 6 and 18 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Scalzi in view of Bharadwaj and U.S. Patent 5,828,884 to Lee et al. (“Lee”).

According, Appellant hereby requests review of each of these grounds of rejection in the present appeal.

VII. Argument

(1) Claims 25-28 and 30-32 comply with 35 U.S.C. § 101:

The final Office Action rejects claims 25-28 and 30-32 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Appellant respectfully disagrees.

Claim 25 recites:

A computer program product for translating binary code instructions from a source format to a target format for processing by a target processor, comprising a computer-readable medium, further comprising:

a template embodied within said computer-readable medium for use in a binary code translator for translating binary code instructions from a source format to a target format for processing by a target processor, the template comprising:

a template identifier for uniquely associating said template to a source instruction;

a set of target instructions in a target format semantically equivalent to the source instruction;

computer usable program code embodied within said computer-readable medium configured to convert said source instruction into a source intermediate data structure having a plurality of members;

computer usable program code embodied within said computer-readable medium configured to map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template; and

computer usable program code embodied within said computer-readable medium configured to convert said target intermediate data structure into a target instruction.

The final Office Action asserts that the computer-readable medium recited in claim 25 “appears to be intended to include a ‘transmission means,’” which, according to the Action, “could be a signal.” (Action, p. 3). To support this assertion, the final Office Action cites to Appellant’s teaching that “any or all of the software used to implement the invention can be communicated via various transmission or storage means such as computer network, floppy disc, CD-ROM or magnetic tape so that the software can be loaded onto one or more devices.” (Appellant’s specification, p. 18, lines 18-20). In doing so, however, the Action ignores that these cited teachings by Appellant are specifically directed to enabling the

software to “be loaded onto one or more devices,” and not specifically to the “computer-readable medium” to which the software is to be loaded and/or stored. (*Id.*, claim 25). It will be readily apparent from the context of claim 25 that the “computer-readable medium” recited in claim 25 refers to a computer-readable medium on which the software is stored, not the transmission means *used to load* the software onto the computer-readable medium on which the software is stored.

Even if, *arguendo*, the “transmission or storage means” taught in the cited portion of Appellant’s specification were referring to the computer-readable medium recited in claim 25, nowhere does Appellant’s specification teach or suggest that the “transmission or storage means such as computer network, floppy disc, CD-ROM or magnetic tape” include or are intended to include “a signal” as the Examiner suggests. In contrast, the specification offers only physical media as examples of the taught “transmission or storage means.” Thus, by insisting that the “computer-readable medium” recited in claim 25 is non-statutory because of the Examiner’s interpretation that the “transmission or storage means” taught in Appellant’s specification “could include a signal,” the Examiner is unfairly reading limitations into Appellant’s claim language that simply are not there and basing the rejection on speculation alone.

With regard to language of claim 25, the MPEP is clear. “When functional descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized.” (MPEP 2106.01).

Claim 25 recites a computer program product having functional descriptive material recorded on “a computer-readable medium.” (Claim 25). Therefore, according to the standards of the MPEP, the “computer-readable medium” recited in claim 25 complies with 35 U.S.C. § 101.

For at least these reasons, the rejection of claim 25 and its dependent claims should not be sustained.

(2) Claims 1-5, 7-17, 19-28, and 30-33 are patentable over Scalzi and Bharadwaj:

Claim 1:

Independent claim 1 recites:

A method of translating *binary* code instructions from a source format to a target format for processing by a target processor, said method comprising the steps of:

identifying a source instruction;

selecting a translation template corresponding to said identified source instruction, said template providing a set of target instructions semantically equivalent to said identified source instruction;

translating said identified source instruction in accordance with said template, wherein said translating comprises:

converting said source instruction into a source intermediate data structure having a plurality of members;

mapping said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template; and

converting said target intermediate data structure into a target instruction; and

outputting said target instruction for processing by said target processor.

(Emphasis added).

Scalzi is directed to “emulation methods for enabling programs written for execution on a computer system built to one type of computer architecture to execute on a target computer system built to a different computer architecture.” (Scalzi, col. 1, lines 11-15).

However, Scalzi does not teach many of the features recited in claim 1. The final Office Action agrees, conceding that:

Scalzi does not specifically teach translating said identified source instruction in accordance with said template, wherein said translating comprises converting said source instruction into a source intermediate data structure having a plurality of members, mapping said members in said source intermediate data structure to

corresponding members in a target intermediate data structure according to said template and converting said target intermediate data structure into a target instruction.

(Action, p. 4).

In light of Scalzi's failure to teach the subject matter of claim 1, the Office Action turns to Bharadwaj, which is directed to a method "for scheduling an instruction of a computer program." (Bharadwaj, abstract). Citing to Bharadwaj's description of a software compiler in which a "parser 38 receives the source code of a program to be compiled as inputs[,]...parses the source language statements[,] and outputs tokenized statements" to an "intermediate representation builder 40" that "receives the tokenized statements as inputs" and "constructs intermediate representations of the tokenized statements," the Action asserts that Bharadwaj teaches "converting said source instruction into a source intermediate data structure having a plurality of members" as recited in claim 1. (Action, pp. 4-5; Bharadwaj, col. 4, lines 1-6; claim 1). Appellant respectfully disagrees.

Appellant reiterates that the process taught in the above cited portion of Bharadwaj is directed to the operation of a software compiler. It is well-known and understood in the art that software compilers transform human-readable source code text into machine-readable binary code. Thus, when Bharadwaj teaches that a parser "receives the source code of a program to be compiled as inputs," "parses the source language statements," and "outputs tokenized statements," Bharadwaj is clearly teaching that text-based source code is parsed as (textual) "language statements" and output as tokenized textual statements to the "intermediate representation builder." (Bharadwaj, col. 4, lines 1-6, emphasis added). Appellant also notes that in contrast to Bharadwaj's teaching, claim 1 explicitly recites that a binary source instruction is identified and translated by converting the binary source instruction into a source intermediate data structure having a plurality of members.

Consequently, where claim 1 requires the conversion of a binary machine-readable source instruction to a binary machine-readable target instruction, Bharadwaj teaches the compiling of a human-readable textual source code instruction into a machine-readable binary instruction. As human-readable textual source code instructions and binary machine-readable instructions are significantly different from each other and not interchangeable, the cited portions of Bharadwaj *cannot* teach the claimed step of “converting said source instruction into a source intermediate data structure having a plurality of members.” (Claim 1). Moreover, nowhere else does Bharadwaj appear to teach or suggest this subject matter.

The final Office Action also asserts that Bharadwaj teaches the step recited in claim 1 of “mapping said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template.” (Action, p. 5). As a basis for this assertion, the Action cites to Bharadwaj’s teaching that the “intermediate representation builder 40 also groups these intermediate representations [of the tokenized source code] into basic blocks and determines how the blocks are interconnected in a control flow diagram.” (*Id.*; Bharadwaj, col. 4, lines 6-12). The final Office Action implicitly contends here that because each of the intermediate representations is added to a basic block, the intermediate representations are “mapped” to “corresponding members in a target intermediate data structure according to said template” as recited in claim 1, wherein the basic blocks are analogous to “the target intermediate data structure.” (*Id.*). Appellant disagrees. The Examiner has not provided any evidence to suggest that the grouping of intermediate representations into basic blocks, as taught by Bharadwaj, is performed “according to [a] template” as recited in claim 1. Bharadwaj does not teach or suggest this subject matter anywhere. To the contrary, by teaching a need for “determin[ing] how the blocks are interconnected in a control flow diagram” and “perform[ing] high-level optimizations on the

“intermediate representations” in the blocks, Bharadwaj appears to teach that the intermediate statements are simply grouped into the basic blocks in the order that they are received by the intermediate representation builder 40. (*Id.* at lines 8-11).

Under the analysis required by *Graham v. John Deere*, 383 U.S. 1 (1966) to support a rejection under § 103, the scope and content of the prior art must first be determined, followed by an assessment of the differences between the prior art and the claim at issue in view of the ordinary skill in the art. The Supreme Court has recently reaffirmed the validity of the *Graham* test. See *KSR International Co. v. Teleflex, Inc.*, 550 U.S. 398, ___ (2007). In the present case, the scope and content of the prior art, as evidenced by Scalzi and Bharadwaj, did not include the claimed subject matter, particularly the steps of “converting said source instruction into a source intermediate data structure having a plurality of members” and “mapping said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template.” (Claim 1).

The differences between the cited prior art and the claimed subject matter are significant because the template-based translation disclosed in claim 1 provides a significant reduction in translation overhead without compromising code quality or flexibility. (Applicant’s specification, p. 17, lines 25-30). Thus, the claimed subject matter provides features and advantages not known or available in the cited prior art. Consequently, the cited prior art will not support a rejection of claim 1 and its corresponding dependent claims under 35 U.S.C. § 103 and *Graham*. For at least these reasons, the rejection of claim 1 and its dependent claims based on Scalzi and Bharadwaj should not be sustained.

Claim 13:

Independent Claim 13 recites:

An apparatus for translating *binary* code instructions from a source format to a target format for processing by a target processor, the apparatus comprising:

- at least one processor configured to execute code embodied on a computer readable medium;
- an instruction identifier embodied within said code for identifying a source instruction;
- a template selector embodied within said code for selecting a translation template corresponding to said identified source instruction, said translation template comprising a set of target instructions semantically equivalent to said identified source instruction and further comprising input and output resources; and
- a translator embodied within said code for translating said identified source instruction, wherein said translator is configured to
 - convert said source instruction into a source intermediate data structure having a plurality of members;*
 - map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template; and*
 - convert said target intermediate data structure into a target instruction; and
 - an output buffer embodied within said code for outputting said target instruction for processing by said target processor.

(Emphasis added.)

In contrast, the combination of Scalzi and Bharadwaj fails to teach or suggest the apparatus of claim 13. As has been amply demonstrated above, Scalzi and Bharadwaj fail to teach or suggest a translator configured to “convert said [binary] source instruction into a source intermediate data structure having a plurality of members” and “map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template.” (Claim 13).

Under the analysis required by *Graham v. John Deere*, 383 U.S. 1 (1966) to support a rejection under § 103, the scope and content of the prior art must first be determined, followed by an assessment of the differences between the prior art and the claim at issue in view of the ordinary skill in the art. The Supreme Court has recently reaffirmed the validity

of the *Graham* test. See *KSR International Co. v. Teleflex, Inc.*, 550 U.S. 398, ____ (2007). In the present case, the scope and content of the prior art, as evidenced by Scalzi and Bharadwaj, did not include the claimed subject matter, particularly a translator configured to “convert said source instruction into a source intermediate data structure having a plurality of members” and “map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template.” (Claim 13).

The differences between the cited prior art and the claimed subject matter are significant because the template-based translation disclosed in claim 13 provides a significant reduction in translation overhead without compromising code quality or flexibility. (Applicant’s specification, p. 17, lines 25-30). Thus, the claimed subject matter provides features and advantages not known or available in the cited prior art. Consequently, the cited prior art will not support a rejection of claim 13 and its corresponding dependent claims under 35 U.S.C. § 103 and *Graham*. For at least these reasons, the rejection of claim 13 and its dependent claims based on Scalzi and Bharadwaj should not be sustained.

Claims 5 and 17:

Claims 5 and 17 recite that “said bit filling routine is uniquely associated with said template.” In contrast, none of the portions of Scalzi cited in the final Office Action as a basis for rejecting claims 5 and 17 teach or suggest that a bit filling routine is uniquely associated with a template. (Action, pp. 5 and 17; Scalzi, col. 18, lines 14-18, col. 19, lines 46-49 and 57-60). For at least this additional reason, the rejection of claims 5 and 17 should not be sustained.

Claim 25:

Independent Claim 25 recites:

A computer program product for translating binary code instructions from a source format to a target format for processing by a target processor, comprising a computer-readable medium, further comprising:

a template embodied within said computer-readable medium for use in a binary code translator for translating binary code instructions from a source format to a target format for processing by a target processor, the template comprising:

a template identifier for uniquely associating said template to a source instruction;

a set of target instructions in a target format semantically equivalent to the source instruction;

computer usable program code embodied within said computer-readable medium configured to convert said source instruction into a source intermediate data structure having a plurality of members;

computer usable program code embodied within said computer-readable medium configured to map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template; and

computer usable program code embodied within said computer-readable medium configured to convert said target intermediate data structure into a target instruction.

(Emphasis added.)

In contrast, the combination of Scalzi and Bharadwaj fails to teach or suggest the computer program product of claim 25. As has been amply demonstrated above, Scalzi and Bharadwaj fail to teach or suggest “computer usable program code...configured to convert said [binary] source instruction into a source intermediate data structure having a plurality of members” and “computer usable program code...configured to map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template.” (Claim 25).

Under the analysis required by *Graham v. John Deere*, 383 U.S. 1 (1966) to support a rejection under § 103, the scope and content of the prior art must first be determined, followed by an assessment of the differences between the prior art and the claim at issue in view of the ordinary skill in the art. The Supreme Court has recently reaffirmed the validity

of the *Graham* test. See *KSR International Co. v. Teleflex, Inc.*, 550 U.S. 398, ____ (2007). In the present case, the scope and content of the prior art, as evidenced by Scalzi and Bharadwaj, did not include the claimed subject matter, particularly “computer usable program code configured to convert said source instruction into a source intermediate data structure having a plurality of members” and “computer usable program code configured to map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template.” (Claim 25).

The differences between the cited prior art and the claimed subject matter are significant because the template-based translation disclosed in claim 25 provides a significant reduction in translation overhead without compromising code quality or flexibility. (Applicant’s specification, p. 17, lines 25-30). Thus, the claimed subject matter provides features and advantages not known or available in the cited prior art. Consequently, the cited prior art will not support a rejection of claim 25 and its corresponding dependent claims under 35 U.S.C. § 103 and *Graham*. For at least these reasons, the rejection of claim 25 and its dependent claims based on Scalzi and Bharadwaj should not be sustained.

Claims 3, 15 and 27:

Claims 3 15, and 27 substantially recite “transform[ing] said data part from said source instruction into said corresponding data part or parts of said set of target instructions.” The final Office Action asserts that Scalzi’s teaching of “the substitution of specific register values, and immediate displacement values, from source machine instructions into target machine instructions which, either singly or as a sequence of more than one, perform the function of the source machine” reads on this subject matter. (Action, pp. 5, 8, and 12; Scalzi, col. 3, lines 2-7). Applicant disagrees. Nowhere in the cited portion does Scalzi teach

or suggest transforming a specific data part or portion of a source instruction into a corresponding specific data part or portion of a target instruction or set of target instructions. Therefore, no *prima facie* case of obviousness has been made. For at least this additional reason, the rejection of claims 3, 15, and 27 should not be sustained.

Claim 33:

Independent Claim 33 recites:

A computer program product for translating *binary* code instructions from a source format to a target format for processing by a target processor, comprising:
a computer-readable medium, comprising:
 a first set of codes for causing a computer to identify a source instruction;
 a second set of codes for causing a computer to select a translation template corresponding to said identified source instruction, said template providing a set of target format instructions semantically equivalent to said identified source instruction;
 a third set of codes for causing a computer to translate said identified source instruction in accordance with said template by converting said identified source instruction to an intermediate source data structure, mapping members of said intermediate source data structure to a members of a target intermediate data structure according to said template, and converting said target intermediate data structure into a target instruction; and
 a fourth set of codes for causing a computer to output said translated instruction for processing by said target processor.

(Emphasis added.)

In contrast, the combination of Scalzi and Bharadwaj fails to teach or suggest the computer program product of claim 33. As has been amply demonstrated above, Scalzi and Bharadwaj fail to teach or suggest “*a third set of codes for causing a computer to translate said identified source instruction in accordance with said template by converting said identified source instruction to an intermediate source data structure, mapping members of said intermediate source data structure to members of a target intermediate data structure*

according to said template, and converting said target intermediate data structure into a target instruction.” (Claim 33).

Under the analysis required by *Graham v. John Deere*, 383 U.S. 1 (1966) to support a rejection under § 103, the scope and content of the prior art must first be determined, followed by an assessment of the differences between the prior art and the claim at issue in view of the ordinary skill in the art. The Supreme Court has recently reaffirmed the validity of the *Graham* test. See *KSR International Co. v. Teleflex, Inc.*, 550 U.S. 398, ___ (2007). In the present case, the scope and content of the prior art, as evidenced by Scalzi and Bharadwaj, did not include the claimed subject matter, particularly “a third set of codes for causing a computer to translate said identified source instruction in accordance with said template by converting said identified source instruction to an intermediate source data structure, mapping members of said intermediate source data structure to members of a target intermediate data structure according to said template, and converting said target intermediate data structure into a target instruction.” (Claim 33).

The differences between the cited prior art and the claimed subject matter are significant because the template-based translation disclosed in claim 33 provides a significant reduction in translation overhead without compromising code quality or flexibility. (Applicant’s specification, p. 17, lines 25-30). Thus, the claimed subject matter provides features and advantages not known or available in the cited prior art. Consequently, the cited prior art will not support a rejection of claim 33 and its corresponding dependent claims under 35 U.S.C. § 103 and *Graham*. For at least these reasons, the rejection of claim 33 and its dependent claims based on Scalzi and Bharadwaj should not be sustained.

(3) Claims 6 and 18 are patentable over Scalzi, Bharadwaj, and Lee:

Claims 6 and 18 depend from independent claims 1 and 13, respectively.

Consequently, claims 6 and 18 are patentable over Scalzi and Bharadwaj for at least the same reasons given above in favor of the patentability of claims 1 and 13.

Moreover, Lee is cited for a teaching of a “method for compiling a software program and executing the program on a system that converts data between little endian and big endian formats” (Lee, Abstract) (Action, p. 15). In contrast, the cited Lee references fail to teach or suggest any of the steps of “converting a source instruction into an intermediate source data structure having a plurality of members,” mapping the members of the intermediate source data structure to “corresponding members in a target intermediate data structure according to said template,” or “converting said target intermediate data structure into a target instruction” that are claimed in claim 6. Similarly, the cited portions of Lee fail to teach or suggest the claimed apparatus having a translator configured to “convert said source instruction into a source intermediate data structure having a plurality of members,” “map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template,” and “convert said target intermediate data structure into a target instruction” that is claimed in claim 18.

Therefore, as demonstrated here, the recent Office Action does not accurately identify the very significant difference between the cited prior art and the claimed subject matter of claims 6 and 18. Moreover, these differences are significant in that the claimed subject matter is not available in the cited prior art. Consequently, for at least the reasons demonstrated here, the cited prior art will not support a rejection of claims 6 and 18 under 35 U.S.C. § 103 and *Graham*. Therefore, the rejection of claims 6 and 18 should not be sustained.

In view of the foregoing, it is submitted that the final rejection of the pending claims is improper and should not be sustained. Therefore, a reversal of the Rejection of October 20, 2008 is respectfully requested.

Respectfully submitted,

DATE: February 18, 2009

/Steven L. Nichols/

Steven L. Nichols
Registration No. 40,326

Steven L. Nichols, Esq.
Managing Partner, Utah Office
Rader Fishman & Grauer PLLC
River Park Corporate Center One
10653 S. River Front Parkway, Suite 150
South Jordan, Utah 84095
(801) 572-8066
(801) 572-7666 (fax)

VIII. CLAIMS APPENDIX

1. (previously presented) A method of translating binary code instructions from a source format to a target format for processing by a target processor, said method comprising the steps of:

identifying a source instruction;

selecting a translation template corresponding to said identified source instruction, said template providing a set of target instructions semantically equivalent to said identified source instruction;

translating said identified source instruction in accordance with said template, wherein said translating comprises:

converting said source instruction into a source intermediate data structure having a plurality of members;

mapping said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template; and

converting said target intermediate data structure into a target instruction; and outputting said target instruction for processing by said target processor.

2. (original) A method according to claim 1 in which said source and target instructions include a control part and a data part and said control part being used in said identification step to identify an instruction.

3. (previously presented) A method according to claim 2 wherein said data part from said source instruction is transformed into said corresponding data part or parts of said set of target instructions.

4. (original) A method according to claim 3 in which said transformation step is carried out in accordance with a bit filling routine associated with said template.

5. (original) A method according to claim 4 in which said bit filling routine is uniquely associated with said template.

6. (original) A method according to claim 3 in which said transformation step is arranged to transform data of one type of endianness to data of another type of endianness.

7. (original) A method according to claim 2 in which said source instruction control parts are each concatenated to provide a unique identifier and said templates are indexed in accordance with said identifiers.

8. (original) A method according to claim 7 in which said templates are indexed by said unique identifiers in a look up table.

9. (original) A method according to claim 1 in which said translation is carried out at runtime of an emulated application program.

10. (original) A method according to claim 1 in which said templates are provided by software procedure calls.

11. (original) A method according to claim 1 in which said source format is 32 bit and said target format is 64 bit.

12. (original) A method according to claim 1 in which said source format is PA-RISC code and said target format is ItaniumTM code.

13. (previously presented) An apparatus for translating binary code instructions from a source format to a target format for processing by a target processor, the apparatus comprising:

at least one processor configured to execute code embodied on a computer readable medium;

an instruction identifier embodied within said code for identifying a source instruction;

a template selector embodied within said code for selecting a translation template corresponding to said identified source instruction, said translation template comprising a set of target instructions semantically equivalent to said identified source instruction and further comprising input and output resources; and

a translator embodied within said code for translating said identified source instruction, wherein said translator is configured to

convert said source instruction into a source intermediate data structure having a plurality of members;

map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template; and

convert said target intermediate data structure into a target instruction; and

an output buffer embodied within said code for outputting said target instruction for processing by said target processor.

14. (original) Apparatus according to claim 13 in which said source and target instructions include a control part and a data part and said instruction identifier uses said control part to identify an instruction.

15. (previously presented) Apparatus according to claim 14 in which said translator is operable to transform said data part from said source instruction into said corresponding data part or parts of said set of target instructions.

16. (original) Apparatus according to claim 15 in which said transformation is carried out in accordance with a bit filling routine associated with said template.

17. (original) Apparatus according to claim 16 in which said bit filling routine is uniquely associated with said template.

18. (previously presented) Apparatus according to claim 15 in which translator is operable to transform data of one type of endianness into data of another type of endianness.

19. (original) Apparatus according to claim 14 in which said source instruction control parts are concatenated to provide a unique identifier and said templates are indexed in accordance with said identifiers.

20. (original) Apparatus according to claim 19 in which said templates are indexed by said unique identifiers in a look up table.

21. (original) Apparatus according to claim 13 in which said translation is carried out at runtime of an emulated application program.

22. (original) Apparatus according to claim 13 in which said templates are provided by software procedure calls.

23. (original) Apparatus according to claim 13 in which said source code has a 32 bit format and said target code has a 64 bit format.

24. (original) Apparatus according to claim 13 in which said source code is PA-RISC code and said target code is ItaniumTM code.

25. (previously presented) A computer program product for translating binary code instructions from a source format to a target format for processing by a target processor, comprising a computer-readable medium, further comprising:

a template embodied within said computer-readable medium for use in a binary code translator for translating binary code instructions from a source format to a target format for processing by a target processor, the template comprising:

a template identifier for uniquely associating said template to a source instruction;

a set of target instructions in a target format semantically equivalent to the source instruction;

computer usable program code embodied within said computer-readable medium configured to convert said source instruction into a source intermediate data structure having a plurality of members;

computer usable program code embodied within said computer-readable medium configured to map said members in said source intermediate data structure to corresponding members in a target intermediate data structure according to said template; and

computer usable program code embodied within said computer-readable medium configured to convert said target intermediate data structure into a target instruction.

26. (previously presented) A computer product according to claim 25, further comprising a set of codes causing a computer to derive the template identifier from a control part of the source instruction.

27. (previously presented) A computer product according to claim 26, wherein the template causes a computer to transform a data part of the source instruction into at least one corresponding data part of the set of target instructions.

28. (previously presented) A computer product according to claim 27, further comprising a set of codes for causing a computer to bit fill the data part of the source instruction.

29. (canceled).

30. (previously presented) A computer product according to claim 26, wherein the template causes a computer to create the template identifier by concatenating the control part of said source instruction.

31. (previously presented) A computer product according to claim 25, wherein the template causing a computer to transform a source instruction having a 32 bit format to a target instruction having a 64 bit format.

32. (previously presented) A computer product according to claim 25, wherein the template causes a computer to transform PA-RISC source code into ItaniumTM target code.

33. (previously presented) A computer program product for translating binary code instructions from a source format to a target format for processing by a target processor, comprising:

a computer-readable medium, comprising:

a first set of codes for causing a computer to identify a source instruction;

a second set of codes for causing a computer to select a translation template corresponding to said identified source instruction, said template providing a set of target format instructions semantically equivalent to said identified source instruction;

a third set of codes for causing a computer to translate said identified source instruction in accordance with said template by converting said identified source instruction to an intermediate source data structure, mapping members of said intermediate source data structure to a members of a target intermediate data structure according to said template, and converting said target intermediate data structure into a target instruction; and

a fourth set of codes for causing a computer to output said translated instruction for processing by said target processor.

34-35. (cancelled).

IX. Evidence Appendix

None

X. Related Proceedings Appendix

None